

# Maxima

## *Część II. Równania*

Zbigniew Koza

Uniwersytet Wrocławski  
Instytut Fizyki Teoretycznej

Wrocław, 2009

# Pierwiastki wielomianów

```
(%i1) allroots(3*x^3+5*x^2+4*x-2);
```

```
(%o1) [x = 0.333333333333333333,
```

```
      x = 1.0 %I - 1.0,
```

```
      x = - 1.0 %I - 1.0]
```

- Funkcja **allroots** znajduje (numerycznie) wszystkie pierwiastki wielomianu rzeczywistego lub zespolonego jednej zmiennej.

# Pierwiastki rzeczywiste

```
(%i1) realroots(x^10+x-1, 1e-7);  
(%o1)      [x = -36096719/33554432,  
            x = 28020603/33554432 ]  
(%i2) %,float;  
(%o2)      [x = -1.075766056776047  
            x = 0.83507904410362]  
(%i3) x^10+x-1, %o1[1], float;  
(%o3)      -1.703474330483346E-7  
(%i4) x^10+x-1, %o1[2], float;  
(%o4)      4.105570328150066E-9
```

- Funkcja **realroots** znajduje (numerycznie) wszystkie rzeczywiste pierwiastki wielomianu rzeczywistego jednej zmiennej.
- **realroots(wielomian, dokładność)**

# Równania liniowe

```
(%i1) x + y = z$
```

```
(%i2) 2*a*x - y = 2*a*a$
```

```
(%i3) y - 2*z = 2$
```

```
(%i4)
```

```
linsolve([%o1,%o2,%o3],[x,y,z]);
```

```
(%o5) [x = a+1, y = 2 a, z = a-1]
```

- Funkcja **linsolve** rozwiązuje układ równań liniowych.

- Składnia:

```
linsolve([równania],[zmienne])
```

# Równania liniowe II

```
(%i1) linsolve([x+y=0,x-y=a],[x,y]);
(%o1)      [x = a/2, y = - a/2];
(%i2) x;
(%o2)      x
(%i3) linsolve([x+y=0,x-y=a],[x,y]),
      globalsolve:true;
(%o3)      [x :  a/2, y :  -a/2]
(%i4) x;
(%o4)      a/2
```

- Parametr **globalsolve** decyduje o tym, czy rozwiązania układ równań mają być podstawiane do zmiennych, względem których rozwiązuje się równania.

# Równania wielomianowe

```
(%i1) e1 : x^2 - y^2 = 3$
```

```
(%i2) e2 : x*y = 2$
```

```
(%i3) algsys([e1,e2], [x,y]);
```

```
(%o3) [[x = -%i, y = 2%i],  
      [x = %i, y = -2%i], [x = 2, y = 1],  
      [x = - 2, y = - 1]]
```

- Polecenie **algsys** służy do rozwiązywania układów równań wielomianów wielu zmiennych

# Równania dowolne I

```
(%i1) solve(asin(cos(2*x))*(x^2-1)
            *(2^x-4) = 0, x);
```

'solve' is using arc-trig functions to get a solution. Some solutions will be lost.

```
(%o1) [x=%pi/4,x=log 4/log
2,x=-1,x=1]
```

- Polecenie **solve** służy rozwiązać dowolne równanie lub układ równań
- **solve(równanie, zmienna)**

# Układy równań dowolnych

```
(%i1) eq1 : x*y^2=x^3-1 $
(%i2) eq2 : x+y=1 $
(%i3) solve ([eq1,eq2],[x,y]);
(%o3)      [[x = -1/2, y = 3/2],
            [x = 1, y = 0]]
```

- **solve** to ogólna funkcja do rozwiązywania równań lub układów równań.
- **solve(expr)** lub **solve(expr = 0)** lub **solve(expr, var)** lub **solve([l\_expr], [l\_var])**



# Równania różniczkowe

```
(%i1) eq: x^2*'diff(y,x) + 3*y*x  
      = cos(x)/x
```

```
(%i2) ode2(eq,y,x);
```

```
(%o2) y=(sin x+%c)/x^3
```

```
(%i9) method;
```

```
(%o9) linear
```

- Polecenie **ode2** rozwiązuje równania różniczkowe zwyczajne stopnia 1 lub 2.
- **%c**, **%k1** i **%k2** symbolizują stałe całkowania.
- Zmienna **method** przechowuje informację o zastosowanej metodzie rozwiązania.
- **ode2(równanie, 'y', 'x')**

# Równania różniczkowe II

```
(%i1) assume(a>0)$
(%i2) depends(y,x)$
(%i3) eq:'diff(y,x,2)=-a*a*y,y,x)$
(%i4) ode2(eq, y, x);
(%o4) y = %k1 sin(a x) + %k2 cos(a x)
(%i5) ic2(%,x=0,y=0,diff(y,x)=1);
(%o5)      y = sin(a x) / a
(%i7)
bc2(%o4,x=0,y=0,x=%pi/(2*a),y=1);
(%o7) y = sin(a x)
```

- **ic2** definiuje warunek początkowy
- **bc2** definiuje warunek brzegowy

# contrib\_ode I

```
(%i1) load(contrib_ode);
```

```
(%o1) C:/PROGRA~1/MAXIMA~1.1/share/maxima/5.9.1/share/
```

```
contrib/diffequations/contrib_ode.mac
```

```
(%i2) eqn: 'diff(y,x)^2+x*'diff(y,x)-y=0;
```

$$(\%o2) \quad \left(\frac{d}{dx}y\right)^2 + x \left(\frac{d}{dx}y\right) - y = 0$$

```
(%i3) contrib_ode(eqn,y,x);
```

First order eq. not linear in y'

```
(%o3) [y=%cx+%c^2,y=-x^2/4]
```

```
(%i4) method;
```

```
(%o4) clairault
```

# contrib\_ode II

## contrib\_ode

- w pierwszej próbie rozwiązuje równanie funkcją **ode2**
- następnie próbuje dodatkowe metody: faktoryzacja, Clairaulta, Lagrange'a, Riccatiego, Abela i Liego.
- czasami podaje rozwiązania w formie parametrycznej
- czasami gubi rozwiązania

## Układy liniowych równań różniczkowych

```
'diff(f(x),x)='diff(g(x),x)+sin(x)$  
'diff(g(x),x,2)='diff(f(x),x)-cos(x)$  
atvalue('diff(g(x),x),x=0,a)$  
atvalue(f(x),x=0,1)  
desolve([%o1,%o2],[f(x),g(x)]);
```

$$[f(x) = a e^x - a + 1, \quad (1)$$

$$g(x) = \cos x + a e^x - a + g(0) - 1] \quad (2)$$

- Polecenie **desolve** rozwiązuje układy liniowych równań różniczkowych przy pomocy transformat Laplace'a.
- Polecenie **atvalue** służy do ustalania wartości początkowych rozwiązań.
- **desolve([równania],[zmienne])**

# Praca z plikami

- Zapisywanie sesji w humanitarnej formie:  
`stringout("d:/plik.mac",input);`
- Odczytywanie sesji:
  - `batch(plik);`
  - `load(plik);`

# Maxima i T<sub>E</sub>X

- Przekształcanie wyrażenia do postaci zrozumiałej dla T<sub>E</sub>X-a:
  - **tex(wyrażenie);**
- Dopisywanie równania do pliku
  - **tex(wyrażenie, nazwa\_pliku);**

To do...

- Grafika
- Macierze
- Programowanie

quit();